



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 10/609,275 | 06/26/2003 | Vinod K. Grover | 3382-65591 | 8309 |

26119 7590 02/22/2006
KLARQUIST SPARKMAN LLP
121 S.W. SALMON STREET
SUITE 1600
PORTLAND, OR 97204

EXAMINER

RUTTEN, JAMES D

ART UNIT PAPER NUMBER

2192

DATE MAILED: 02/22/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

| | | | |
|------------------------------|------------------------|---------------------|--|
| Office Action Summary | Application No. | Applicant(s) | |
| | 10/609,275 | GROVER ET AL. | |
| | Examiner | Art Unit | |
| | J. Derek Rutten | 2192 | |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 December 2005.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-13, 15-23 and 25-55 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-12, 15-23 and 25-54 is/are rejected.
- 7) ☒ Claim(s) 13 and 55 is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input checked="" type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date <u>12/4/05</u> | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to Applicant's amendment dated 12/9/2005, responding to the 7/14/2005 Office action provided in the rejection of claims 1-53, wherein claims 1, 2, 5, 9, 11, 13, 15-18, 23, 25, 27, 29, 32, 44, 46-48 and 50 have been amended, claims 14 and 24 have been canceled, and new claims 54 and 55 have been added. Claims 1-13, 15-23, and 25-55 remain pending in the application and have been fully considered by the examiner.

2. Applicant's arguments, see page 19, filed 12/9/2005, with respect to the rejection(s) of claim 1 under 35 U.S.C. 102(b) have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made in view of "Component Design of Retargetable Program Analysis Tools that Reuse Intermediate Representations" by Hayes et al.

3. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Response to Amendments/Arguments

4. Applicant's amendments have overcome the objections to the specification and claims.

Therefore, these objections have been withdrawn.

5. Applicants amendments to claims 1, 15, 32, 44, 46-48, and 50 have overcome the rejections under 35 USC §§ 101 and 112, second paragraph. Therefore, these rejections have been withdrawn.

6. On page 19, Applicant essentially argues that Burke's intermediate representation is specific to the Java programming language. Similar arguments are presented on pages 26 and 29-31. While Burke does not appear to expressly teach dependence upon the source language, neither does it particularly teach independence. As such, these arguments are convincing. However, a new rejection is made in view of "Component Design of Retargetable Program Analysis Tools that Reuse Intermediate Representations" by Hayes et al.

7. On page 21 Applicant essentially argues that Burke does not explicitly express exception handling control flow as recited by claim 1. Similar arguments are presented on page 23 with respect to claim 53. However, the mere presence of an explicit exception handling instruction provides some level of control flow. Exception handling by its very nature is intertwined with control flow. Upon the generation of an exception, program flow is diverted to an exception handler. The presence of an explicit exception handling instruction therefore provides for control flow as well. Further discussion of Burke's HIR is found in prior art of record "Efficient

Art Unit: 2192

and precise modeling of exceptions for the analysis of Java programs” by Choi et al. Figure 2 on page 22 shows a segment of intermediate code including the exception handling code discussed in Burke. Handling of the instructions “null_check” and “bounds_ckeck” are provided by exception handlers “java.lang.NullPointerException handler” and “java.lang.Exception handler” and clearly provide exception handling control flow in the intermediate representation.

Therefore, Applicant’s argument is not convincing.

8. In response to applicant's arguments at the top of page 24, the recitation “multiple source code languages” has not been given patentable weight because the recitation occurs in the preamble. A preamble is generally not accorded any patentable weight where it merely recites the purpose of a process or the intended use of a structure, and where the body of the claim does not depend on the preamble for completeness but, instead, the process steps or structural limitations are able to stand alone. See *In re Hirao*, 535 F.2d 67, 190 USPQ 15 (CCPA 1976) and *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951).

Claim Rejections - 35 USC § 103

9. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

10. Claims 1, 15, 16, and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over “The Jalapeno dynamic optimizing compiler for Java” by Burke et al. (hereinafter “Burke”) in

Art Unit: 2192

view of “Component Design of Retargetable Program Analysis Tools that Reuse Intermediate Representations” by Hayes et al. (hereinafter “Hayes”)

In regard to claim 1, Burke discloses:

A method implemented at least in part by a computing device of processing a ...intermediate representation of software comprising exception handling constructs, the method comprising:

reading the ...intermediate representation of software comprising exception handling constructs; wherein the ...intermediate representation explicitly expresses exception handling control flow of the software; See page 131, paragraph 2 of section 4:

A key difference between the Jalapeno HIR and Java bytecodes is the addition of separate operators to implement explicit checks for several common run-time exceptions, e.g., NULL-CHECK and BOUNDS-CHECK operators to test for null pointer dereferences and out-of-bounds array accesses respectively.

and generating, in a computer-readable media having a tangible component, a computer-readable version of the software implementing the exception handling control flow based on the ... intermediate representation. See bottom of Figure 3: “Binary Code”.

Burke does not expressly disclose a *source language independent* intermediate representation. However, in an analogous environment, Hayes teaches a source language independent intermediate representation. See page 363, section 6 column 2:

Retargetable compilers ease the process of adapting to both different languages and different machine architectures by defining a common intermediate program representation. Language-specific translators transform a source program into this common representation, then pass the result to language-independent components for additional translation.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Hayes' teaching of a source language independent intermediate language with Burke's intermediate representation. One of ordinary skill would have been motivated to provide a language independent representation in order reuse complex optimization modules created for the representation (see Hayes top of column 1 on page 364).

In regard to claim 15, Burke discloses a system. See the abstract. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 16, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claim 1.

In regard to claim 53, all limitations have been addressed in the above rejection of claim 15.

11. Claims 2, 5-7, 17, 18, 21, 22, 32-36, 39 and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke as applied to claims 1, 15, 16, and 53 above, and further in view of prior art of record "Compilers: Principles, Techniques, and Tools" by Aho et al. (hereinafter "Aho"), further in view of "Pure Java 2" by Litwak (hereinafter "Litwak") further in view of prior art of record "Marmot: An Optimizing Compiler for Java" by Fitzgerald et al. (hereinafter "Fitzgerald"), further in view of US 6,421,667 B1 to Codd et al. (hereinafter "Codd").

In regard to claim 2, Burke does not expressly disclose instructions related to finalization code blocks. However, in an analogous environment, Aho teaches that instructions are used to transfer control in intermediate representations. See page 491, Example 8.5: “goto”. Further, Litwak teaches that a finalization block is executed when it is present. See the top of page 3:

Additionally, you might have a finally clause, in which you perform code that you want done all the time, whether you get an exception or not.

Further, Fitzgerald discusses requirements for blocks of an intermediate representation.

See page 3, paragraphs 2 and 4 in section 3.1:

A method is represented as a control flow graph with a distinguished root (entry) block. Each graph node (basic block) consists of a sequence of effect statements and concludes with a control statement. An effect statement is either a side effect statement or an assignment statement. A side effect consists of an expression, and represents a statement that does not record the result of evaluating the expression. **Each basic block concludes with a control statement that specifies the succeeding basic block**, if any, to execute under normal control flow.

JIR models Java exception handling by labeling basic blocks with distinguished exception edges. These edges indicate the class of the exceptions handled, the bound variable in the handler, and **the basic block to transfer control to** if that exception occurs during execution of the guarded statements. The presence of an exception edge does not imply that the block will throw such an exception under some execution path.

Further still, Codd teaches that initialization code can be used at the beginning of a block to accept a transfer of control. See column 18 lines 39-41:

The queue monitor initialization code in logic block 875 is principally invoked whenever a new entry is placed into the DTQ table as represented by logic block 871.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho’s instructions with Litwak’s teaching of finalization blocks with Fitzgerald’s flow control with Codd’s teaching of control acceptance with Burke’s

intermediate representation. One of ordinary skill would have been motivated to provide flow control in order to correctly implement a finalization block.

In regard to claim 5, Burke does not expressly disclose: *wherein destination operands of the second instruction are the same as source operands of the third instruction*. However, Aho teaches that the outputs of an instruction can be used as the input for another instruction. See page 592, Fig. 10.6: $t_6 := 4 * i \dots x := a[t_6]$. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho's teaching of dependent instructions with Burke's intermediate representations. One of ordinary skill would have been motivated to provide data dependence between instructions in order to use the results of one computation as the basis for another.

In regard to claim 6, the above rejection of claim 2 is incorporated. Burke discloses the use of labels throughout, e.g. Figure 6. However, Burke does not expressly describe labels in the context of flow control. However, Aho teaches that labels are an elementary programming language construct. See page 506: *Labels and Gotos*. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho's teaching of labels to implement Burke's flow control. One of ordinary skill would have been motivated to change the flow of a program using an intuitive construct.

In regard to claim 7, the above rejection of claim 2 is incorporated. All further limitations have been addressed in the above rejection of claim 6.

In regard to claims 17, 18, 21, and 22, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 2, and 5-7, respectively.

In regard to claim 32, Burke does not expressly disclose a storage medium. However, Cobb teaches a storage medium. See column 5 lines 39-46. All further limitations have been addressed in the above rejection of claim 2.

In regard to claims 33-36 and 39, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejection of claims 1 and 5-7.

In regard to claim 40, the above rejection of claim 32 is incorporated. Burke does not expressly disclose: *wherein control flow to the finalization block is expressed by a related set of FINAL, FINALLY and ENDFINALLY instructions*. However, Aho discloses that instructions can be used to describe flow control with intermediate representations. See page 468, paragraph 2. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho's teaching of language design with Burke's intermediate representation. One of ordinary skill would have been motivated to define instructions that guide the flow of control in order to inform a back end compiler.

Art Unit: 2192

12. Claims 3, 4, 19, 20, 41, and 42 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke, Aho, Litwak, Fitzgerald, and Codd as applied to claims 2, 5-7, 17, 18, 21, 22, 32, 33-36, 39 and 40 above, and further in view of US 5,918,235 to Kirshenbaum et al. (hereinafter “Kirshenbaum”).

In regard to claim 3, the above rejection of claim 2 is incorporated. Burke does not expressly disclose: *wherein the finalization code block comprises code related to destructor of an object*. However, in an analogous environment, Kirshenbaum teaches that a finalization routine acts as a destructor. See column 7 lines 50-64. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Kirshenbaum’s teaching of destructors with Litwak’s teaching of finalization routines. One of ordinary skill would have been motivated to guarantee that an object is destroyed by placing it in the finalization block.

In regard to claim 4, all further limitations have been addressed in the above rejection of claim 3.

In regard to claims 19 and 20, the above rejection of claim 17 is incorporated. All further limitations have been addressed in the above rejection of claims 3 and 4, respectively.

In regard to claims 41 and 42, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejection of claims 3 and 4.

13. Claims 8, 37 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke, Aho, Litwak, Fitzgerald, and Codd as applied to claims 2, 5-7, 17, 18, 21, 22, 32, 33-36, 39 and 40 above, and further in view of US 5,966,702 to Fresko et al. (hereinafter "Fresko").

In regard to claim 8, the above rejection of claim 2 is incorporated. Burke does not expressly disclose *wherein the third instruction for expressing transfer of control out of the finalization code block comprises fields for indicating different continuations for control transfer out of the finalization code block based on whether entry into the finalization code block was explicit or due to an exception*. However, in an analogous environment, Fresko teaches that a continuation out of a finalization code block requires an indication of a return based upon how the finalization block was reached. See column 42 lines 10-39. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Fresko's teaching of continuation with Burke's intermediate representation. One of ordinary skill would have been motivated to return to the proper block of code upon execution of a *finally* subroutine.

In regard to claim 37, the above rejection of claim 32 is incorporated. All further limitations have been addressed in the above rejection of claim 8.

In regard to claim 38, the above rejection of claim 37 is incorporated. All further limitations have been addressed in the above rejection of claim 23.

14. Claims 9, 10, 25, and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke as applied to claims 1, 15, 16, and 53 above, and further in view of Aho and Litwak.

In regard to claim 9, the above rejection of claim 1 is incorporated. Burke does not expressly disclose *catching an exception, returning an exception object related to the exception or specifying a handler for the exception based on a type value of the exception object*. However, in an analogous environment, Litwak teaches catching exceptions, returning exception objects, and specifying handlers based on type. See page 3:

Java processes catch blocks by examining the type of the exception class listed in the catch block parameter list. The catch block whose Exception type matches the type of Exception thrown from the try block is executed.

All further limitations have been addressed in the above rejection of claim 2. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Litwak's teaching of catching exceptions with Aho's teaching of intermediate representation instructions with Burke's intermediate representation. One of ordinary skill would have been motivated to correctly implement exception handling using a try-catch mechanism.

In regard to claim 10, the above rejection of claim 9 is incorporated. Burke does not expressly disclose: *wherein the second instruction for specifying the handler comprises: at least one Boolean source operand for indicating the type value of the*

exception object; at least one source operand indicating a label preceding a code block related to the handler to which control flow will pass if the Boolean source operand is true; and at least one source operand indicating a label preceding a code block related to a continuation to which control flow will pass if the Boolean source operand is false.

However, Aho teaches that a conditional jump can be used to redirect program flow in the presence of a binary relation. See page 467, item 5. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Aho's teaching of conditional jumps with Burke's intermediate representation. One of ordinary skill would have been motivated to represent control flow in a single operation in order to save space and enhance program understanding.

In regard to claims 25 and 26, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 9 and 10, respectively.

15. Claims 11, 12, 27, 28, 48, and 49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burk as applied to claims 1, 15, 16, and 53 above, and further in view of Aho, Litwak and Fresko.

In regard to claim 11, the above rejection of claim 1 is incorporated. Burke does not expressly disclose: *wherein the uniform intermediate representation comprises: an instruction for specifying a handler for an exception based on a type value of an*

exception object related to the exception, wherein a destination operand of the instruction comprises a predetermined exception object, a first source operand of the instruction comprises a label indicative of a code block related to the handler and second source operand comprises a label indicative of a code block related to a continuation.

However, Aho teaches that instructions are used to transfer control in intermediate representations. See page 491, Example 8.5: “goto”. Litwak teaches that handler destinations are chosen based on the type of an exception object. See page 4 paragraph 2. Further, Fresko teaches that continuations are required to provide a return to code execution. See column 42 lines 10-39. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Litwak’s teaching of handlers with Fresko’s teaching of continuations with Aho’s teaching of instructions with Burke’s intermediate representation. One of ordinary skill would have been motivated to provide instructions for representation of control flow.

In regard to claim 12, the above rejection of claim 11 is incorporated. Fresko teaches that a determination of exception type results in the execution of a handler, or in a continuation. See column 42 lines 5-39. All further limitations have been addressed in the above rejection of claim 10.

In regard to claims 27 and 28, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claims 11 and 12.

In regard to claims 48 and 49, all limitations have been addressed in the above rejection of claims 11 and 12.

16. Claim 13, 29-31, and 50-52 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke as applied to claims 1, 15, 16, and 53 above, and further in view of Aho, Litwak, Fresko, and "C/C++ Language Reference: try-except Statement", by Microsoft (hereinafter "Microsoft Language Reference").

In regard to claim 13, the above rejection of claim 1 is incorporated. Burke does not expressly disclose: *wherein the source language independent intermediate representation comprises: a first instruction for indicating entry into a try-except region; and a second instruction for selecting one of a plurality of control flow paths for exception handling based on a type value related to the exception, wherein the plurality of control flow paths available for selection includes a path related to resumption of execution of an instruction causing the exception;* However, in an analogous environment, the Microsoft Language Reference teaches that try-except statements are a custom extension to a standardized language. See page 1 paragraph 1. Further, Aho teaches that instructions are used to transfer control in intermediate representations. See page 491, Example 8.5: "goto". Further, Litwak teaches that if an exception is raised, then a program should continue execution after processing the exception. See middle of page 5. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use the teaching of Microsoft Language Reference with Aho's

teaching of instructions with Litwak's teaching of exception flow with Burke's intermediate representation. One of ordinary skill would have been motivated to implement a programming language feature efficiently in an intermediate representation to assist in compilation. Further, one would be motivated to provide continued execution after an exception is raised. All further limitations have been addressed in the above rejection of claim 11.

In regard to claim 30, the following rejection of claim 29 is incorporated. All further limitations have been addressed in the above rejection of claim 13.

In regard to claim 31, the above rejection of claim 30 is incorporated. Burke does not expressly disclose: *wherein a handler for the first instruction for indicating entry into the try-except region is the same as a handler for the exception causing instruction.* However, Litwak teaches that the same handler can handle different exceptions. See bottom of page 4. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Litwak's teaching of exception handlers with Burke's teaching of intermediate representations. One of ordinary skill would have been motivated to use previously implemented code to handle multiple exceptions in order to reduce coding time.

In regard to claims 51 and 52, the following rejection of claim 50 is incorporated. All limitations have been addressed in the above rejections of claims 13 and 31.

17. Claim 29 and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke as applied to claims 1, 15, 16, and 53 above, and further in view of Aho, Litwak, and Microsoft Language Reference.

In regard to claim 29, the above rejection of claim 15 is incorporated. All further limitations have been addressed in the above rejection of claim 13.

In regard to claim 50, all limitations have been addressed in the above rejection of claim 13.

18. Claim 43 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burke, Aho, Litwak, Fitzgerald, and Codd as applied to claims 3, 4, 19, 20, 41, and 42 above, and further in view of US 6289446 B1 to Nilsson (hereinafter "Nilsson").

In regard to claim 43, the above rejection of claim 42 is incorporated. Burke does not expressly disclose: *wherein the expression temporary object is created upon a condition being true and the control is transferred to the finalization code block upon the same condition being true*. However, in an analogous environment, Nilsson teaches that temporary objects are created during creation of an exception. See column 17 line 63 – column 18 line 13. Litwak teaches that a finalization block is entered for *try-finally* styled exception processing. See top of page 3. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Nilsson's teaching of

Art Unit: 2192

temporary object creation with Litwak's teaching of finalization. One of ordinary skill would have been motivated to create temporary objects for programming languages that promote their use.

19. Claims 44-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Burke and further in view of Aho, Litwak, and Codd.

In regard to claim 44, all limitations have been addressed in the above rejections of claims 9 and 32.

In regard to claim 45, the above rejection of claim 44 is incorporated. All further limitations have been addressed in the above rejection of claim 10.

In regard to claim 46, the above rejection of claim 45 is incorporated. Burke does not expressly disclose: *wherein the code block related to the continuation is related to a filter*. However, Litwak teaches that after a filter determines what handler to use, it forwards execution. See top of page 3. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Litwak's continuation with Burke's intermediate representation. One of ordinary skill would have been motivated to process an exception through a filter to determine where processing should continue.

Art Unit: 2192

20. Claim 47 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burke, Aho, Litwak, and Codd as applied to claim 45 above, and further in view of “A single intermediate language that supports multiple implementations of exceptions” by Ramsey et al. (hereinafter “Ramsey”).

In regard to claim 47, the above rejection of claim 45 is incorporated. Burke does not expressly disclose: *wherein the code block related to the continuation comprises an unwind instruction*. However, in an analogous environment, Ramsey teaches that unwind instructions are called to find a handler that inherently provides continuations. See page 286, column 1. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Ramsey’s teaching of unwind instructions with Burke’s intermediate representations. One of ordinary skill would have been motivated to unwind a stack in order to find an exception handler.

21. Claim 54 is rejected under 35 U.S.C. 103(a) as being unpatentable over Burke and Hayes as applied to claim 1 above, and further in view of U.S. Patent 6,253,306 to Ben-Meir et al. (hereinafter “Ben-Meir”) further in view of U.S. Patent 4,197,578 to Wada et al. (hereinafter “Wada”).

In regard to claim 54, the above rejection of claim 1 is incorporated. Burke does not expressly disclose exception causing instructions with labels representing their handlers wherein each instruction comprises a handler field. However, in an analogous

environment, Ben-Meir teaches an exception instruction with a label representing an exception handler. See column 10 lines 34-49. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Ben-Meir's label with Burke's instruction. One of ordinary skill would have been motivated to provide a label in order to enhance readability of code. Burke does not expressly disclose wherein each instruction comprises a handler field. However, in an analogous environment, Wada teaches that every instruction can be made up of a field to designate an operand. See FIG. 2 and column 4 lines 49-53. It would have been obvious to one of ordinary skill in the art at the time the invention was made to use Wada's field with Burke's handler. One of ordinary skill would have been motivated to provide a standard format in order to simplify processing.

Allowable Subject Matter

22. Claims 13, 23, and 55 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

23. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. U.S. Patent Application Publication No. 2004/0025152 A1 to Ishizaki et al. discloses an intermediate language representation including an exception causing instruction with an explicit exception handler field. See Fig. 1.


Art Unit: 2192

24. Any inquiry concerning this communication or earlier communications from the examiner should be directed to J. Derek Rutten whose telephone number is (571) 272-3703. The examiner can normally be reached on T-F 6:00 - 4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

jdr



TUAN DAM
SUPERVISORY PATENT EXAMINER